CLAIMS

For the convenience of the Examiner, all claims have been presented whether or not an amendment has been made. The claims have been amended as follows:

1. **(Currently Amended)** A method of detecting polymorphic viral code ~~in a computer program~~, comprising~~:~~ ~~the steps of:~~

~~(a)~~ emulating a first predetermined number of instructions of ~~the~~ **a** computer program;

detecting at least one unused or misused operand or operator of the first predetermined number of instructions;

~~(b)~~ collecting information corresponding to ~~a state of~~ a plurality of registers and/or flags after **emulating at least one instruction;** ~~each emulated instruction execution;~~ and

~~(c)~~ determining a probability that the computer program contains polymorphic viral code based **at least in part** on an heuristic analysis of the collected ~~register/flag state~~ information.

2. **(Currently Amended)** The method of claim 1, further comprising emulating ~~a second predetermined number of~~ **additional** instructions if the ~~probability~~ determined **probability** ~~in step (c)~~ is above a predetermined threshold**.** ~~, wherein the second predetermined number of instructions is greater than the first predetermined number of instructions.~~

3. **(Currently Amended)** The method of claim 2, wherein the ~~second predetermined number of~~ **additional** instructions ~~corresponds~~ **correspond** to execution of a polymorphic decryptor.

4. **(Currently Amended)** The method of claim 1, further comprising ~~monitoring~~ **detecting improper use of at least one of** the plurality of registers and/or flags**.** ~~for improper register/flag usage.~~

5. **(Currently Amended)** The method of claim 4, further comprising ~~maintaining~~ **determining**, for each of the plurality of registers and/or flags, ~~a corresponding count of~~ a number of times that the ~~register/flag~~ **register and/or flag** was improperly used during the emulation of **the first predetermined number of instructions.** ~~instructions in step (a).~~

6. **(Canceled)**

7. **(Currently Amended)** The method of claim ~~6,~~ **1, wherein detecting at least one unused or misused operand or operator comprises identifying at least one operand or operator that is not used during emulation of the first predetermined number of instructions.** ~~further comprising detecting when an operand value of an instruction which is set is not used by the instruction.~~

8. **(Currently Amended)** The method of claim ~~6,~~ **1, wherein detecting at least one unused or misused operand or operator comprises identifying at least one undefined operand or operator used during emulation of the first predetermined number of instructions.** ~~further comprising detecting when an undefined operand of an instruction is used by the instruction.~~

9. **(Currently Amended)** A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform **a** method ~~steps~~ for detecting polymorphic viral code in a subject computer program, the method ~~steps~~ comprising:

emulating a selected number of instructions of the computer program;

detecting at least one unused or misused operand or operator of the selected number of instructions;

collecting information corresponding to ~~a state of~~ a plurality of ~~registers/flags~~ **registers and/or flags** after ~~each emulated instruction execution~~ **emulating at least one instruction**; and

determining a probability that the computer program contains polymorphic viral code based **at least in part** on an heuristic analysis of the collected ~~register/flag state~~ information.

10. **(Currently Amended)** A ~~computer~~ system **for detecting polymorphic viral code**, comprising:

a processor; and

a program storage device readable by the ~~computer~~ system, tangibly embodying a program of instructions executable by the processor to perform **a** method ~~steps~~ for detecting a polymorphic viral code in a ~~subject~~ computer program, the method ~~steps~~ comprising:

emulating a selected number of instructions of the computer program;

detecting at least one unused or misused operand or operator of the selected number of instructions;

collecting and storing information corresponding to ~~a state of~~ a plurality of ~~registers/flags~~ **registers and/or flags** after ~~each emulated instruction execution~~ **emulating at least one instruction**; and

determining a probability that the computer program contains polymorphic viral code based **at least in part** on an heuristic analysis of the collected ~~register/flag state~~ information.

11. **(Currently Amended)** A computer data signal embodied in a transmission medium**, the computer data signal embodying** ~~which embodies~~ instructions executable by a computer to detect polymorphic viral code in a computer program**, the computer data signal** comprising:

a first segment including emulator code to emulate a selected number of instructions of the computer program;

a second segment including detection code to detect at least one unused or misused operand or operator of the selected number of instructions;

a ~~second~~ **third** segment including analyzer code to analyze a plurality of ~~registers/flags~~ **registers and/or flags** accessed during emulation of ~~the instructions~~ **at least one instruction**; and

a ~~third~~ **fourth** segment including heuristic processor code to determine a probability that the computer program contains polymorphic viral code based **at least in part** on an heuristic analysis of **the plurality of registers and/or flags.** ~~register/flag state information supplied by the analyzer code.~~

12. **(Currently Amended)** An apparatus for detecting polymorphic viral code ~~in a computer program~~, comprising:

an emulator **operable to emulate**~~, wherein the emulator emulates~~ a first predetermined number of instructions of ~~the~~ **a** computer program;

an operational code analyzer ~~that analyzes~~ **operable to:**

detect at least one unused or misused operand or operator of the first predetermined number of instructions; and

analyze a plurality of ~~registers/flags~~ **registers and/or flags** accessed during emulation of ~~the instructions~~ **at least one instruction**;

and

an heuristic analyzer~~, wherein the heuristic analyzer determines~~ **operable to determine** a probability that the computer program contains polymorphic viral code based **at least in part** on an heuristic analysis of **the plurality of registers and/or flags.** ~~register/flag state information supplied by the operational code analyzer.~~

13. **(Currently Amended)** The apparatus of claim 12, wherein the emulator **is operable to** emulates ~~a second predetermined number of~~ **additional** instructions if the **determined** probability ~~determined by the heuristic analyzer~~ is above a predetermined threshold**.** ~~, the second predetermined number of instructions being greater than the first predetermined number of instructions.~~

14. **(Currently Amended)** The apparatus of claim 13, wherein the ~~second predetermined number of~~ **additional** instructions corresponds to execution of a polymorphic decryptor.

15. **(Currently Amended)** The apparatus of claim 12, wherein the operational code analyzer **is operable to detect improper use of at least one of** ~~monitors~~ the plurality of registers and/or flags. ~~for improper register/flag usage.~~

16. **(Currently Amended)** The apparatus of claim 15, wherein the heuristic analyzer ~~maintains~~ **determines**, for each of the plurality of registers and/or flags, ~~a~~

~~corresponding count of~~ a number of times that the ~~register/flag~~ **register and/or flag** was improperly used during the **emulation of the first predetermined number of instructions.** ~~emulated instructions.~~

17. **(Canceled)**

18. **(Currently Amended)** The apparatus of claim ~~17~~ **12**, wherein **detecting at least one unused or misused operand or operator comprises identifying at least one undefined operand or operator used during emulation of the first predetermined number of instructions.** ~~the operational code analyzer detects when an operand value of an instruction which is set is not used by the instructions.~~

19. **(Currently Amended)** The apparatus of claim ~~17~~ **12**, wherein **detecting at least one unused or misused operand or operator comprises identifying at least one undefined operand or operator used during emulation of the first predetermined number of instructions.** ~~the operational code analyzer detects when an undefined operand of an instruction is used by the instruction.~~

20. **(New)** The method of claim 1, further comprising:

identifying a polymorphic viral code associated with the computer program; and

generating or modifying at least one rule based at least in part on the identification of the polymorphic viral code.

21. **(New)** The method of claim 1, wherein the heuristic analysis is performed using at least one neural network.

22. **(New)** The method of claim 2, wherein the additional instructions are more than the first predetermined number of instructions.

23. **(New)** The method of claim 5, wherein the heuristic analysis comprises comparing the number of times that the register and/or flag was improperly used with the first predetermined number of instructions.

24. **(New)** The method of claim 5, wherein the heuristic analysis comprises comparing the number of time that the register and/or flag was improperly used with statistics corresponding to a plurality of polymorphic viral codes.

25. **(New)** The method of claim 7, wherein the determined probability is based at least in part on the identification of the at least one unused operand or operator.

26. **(New)** The method of claim 8, wherein the determined probability is based at least in part on the identification of the at least one undefined operand or operator.

27. **(New)** The apparatus of claim 12, wherein at least one of the emulator, operational code analyzer, or heuristic analyzer is further operable to:

identify a polymorphic viral code associated with the computer program; and

generate or modifying at least one rule based at least in part on the identification of the polymorphic viral code.

28. **(New)** The apparatus of claim 12, wherein the heuristic analysis is performed using at least one neural network.

29. **(New)** The apparatus of claim 13, wherein the additional instructions are more than the first predetermined number of instructions.

30. **(New)** The method of claim 16, wherein the heuristic analysis comprises comparing the number of times that the register and/or flag was improperly used with the first predetermined number of instructions.

31. **(New)** The method of claim 16, wherein the heuristic analysis comprises comparing the number of time that the register and/or flag was improperly used with statistics corresponding to a plurality of polymorphic viral codes.

32. **(New)** The method of claim 18, wherein the determined probability is based at least in part on the identification of the at least one unused operand or operator.

33. **(New)** The method of claim 19, wherein the determined probability is based at least in part on the identification of the at least one undefined operand or operator.